

Fundamentals of Computing II
S2020
CSCE110101/02
Assignment 1
Revision on CSCE1001

Before starting your assignment, you should study/revise and implement the material you studied in the CSCE1001 curriculum, stressing on the following topics/techniques:

- **Top down design (modular programming).**
 - **Recursive functions.**
 - **Object oriented programming (simple classes with overloaded constructors).**
 - **Data structures (1D/2D) static arrays.**
 - **Stream operations (reading input/writing output from/to file streams as well as from/to standard input/output devices).**
-

Problem Statement:

It is required to design and implement a simple board game where some farm animals (mouse, turtle, hen, etc.) are on an island with 2 cats. All animals walk/run on the ground (no flying animals). Each animal will be given a set of directions to move around the board.

After you run your program, it is required to output the result of the movement of every farm animal as one of the following:

- "Eaten by the cat."
- "Escaped through the bridge."
- "Drowned outside the island."
- "Starved... Stuck inside the board."

Technical Specifications:

A- For each animal:

Attributes: (private)

- rowPosition (integer)
- columnPosition (integer)
- name (string)

Methods (public):

i- Constructors:

- a. default constructor → sets the rowPosition and columnPosition to 0.
- b. constructor → takes two parameters to set rowPosition and columnPosition.

- ii- Modifiers:
 - a. move(char c) Implementation → updates rowPosition or columnPosition based on the character c which can have any of the values ('U' (up), 'D' (down), 'L' (left), and 'R' (right)). The function returns false if the movement in the specified direction will lead to going beyond the bounds of the board, otherwise, it returns true.
- iii- Getters:
 - a. getRow() Implementation
 - b. getColumn() Implementation

B- For the board:

Attributes: (private)

- boardSize (integer)
- Board (2D character array)

Methods (private):

- i- Constructor:

takes the following integer parameters (int c1r, int c1c, int c2r, int c2c) then: initializes the array to be boardSize x boardSize, initializes the whole board array to the value '-' except for three cells:

 - 2 cells will contain the value 'C' representing the 2 cats at row c1r and column c1c, and at row c2c and column c2c respectively.
 - 1 cell will contain the value 'B' representing the bridge to the exit, which is always at the last column and the middle row.
- ii- Getter:
 - a. getBoardCell(int r, int c) → returns the char value inside the cell at row r and column c on the board array.

Game Description:

It is required to design a main function that performs the following operations:

- Constructs the board, initializing the boardSize attribute to number entered by the user (between 9 and 12) and sets two positions (correct positions) for the 2 cats.
- Reads from the user the number of farm animals to wonder on the board. For each farm animal, it is required to enter initial data using one of the constructors.
- Reads from an input file a number of strings equivalent to the number of farm animals, we call this the movements' string of the farm animal. Each string will consist of the characters an object will move, for example "UDLD". The length of each string is 4 characters.

The game is based on an iterative scenario through each string to deduce the final state/destination of each object. Based on the result of the movement, the program prints one of the messages below:

- "Eaten by the cat." if the farm animal encounters any cat.

- "Escaped through the bridge." if the farm animal encounters the bridge.
 - "Drowned outside the island." if the farm animal moved out of the board.
 - "Starved... Stuck inside the board." otherwise, i.e. if the farm animal is still on the board.
- Once a farm animal escapes, gets eaten or drowns you should not continue with the processing of its movements' string.

Sample Runs:

Assume the user entered 9 for the boardSize.

Assume that the positions of the cats are (3, 5) and (4, 1).

Assume the user chose to have 3 farm animals; Duck, Mouse and Turtle.

Assume that the initial position of the "Duck" is (1, 1), and that of the "Mouse" is (2, 2), and that of the "Turtle" is (5, 5).

Run1:

If the main function reads the following movements' strings:

"DDDD"

"RRUL"

"URRR"

The output will be:

Duck: Eaten by the cat.

Mouse: Starved... Stuck inside the board.

Turtle: Escaped through the bridge.

Run2:

If the main function reads the following movements' strings:

"UURR"

"RRRD"

"LDRL"

The output will be:

Duck: Drowned outside the island.

Mouse: Eaten by the cat.

Turtle: Starved... Stuck inside the board.